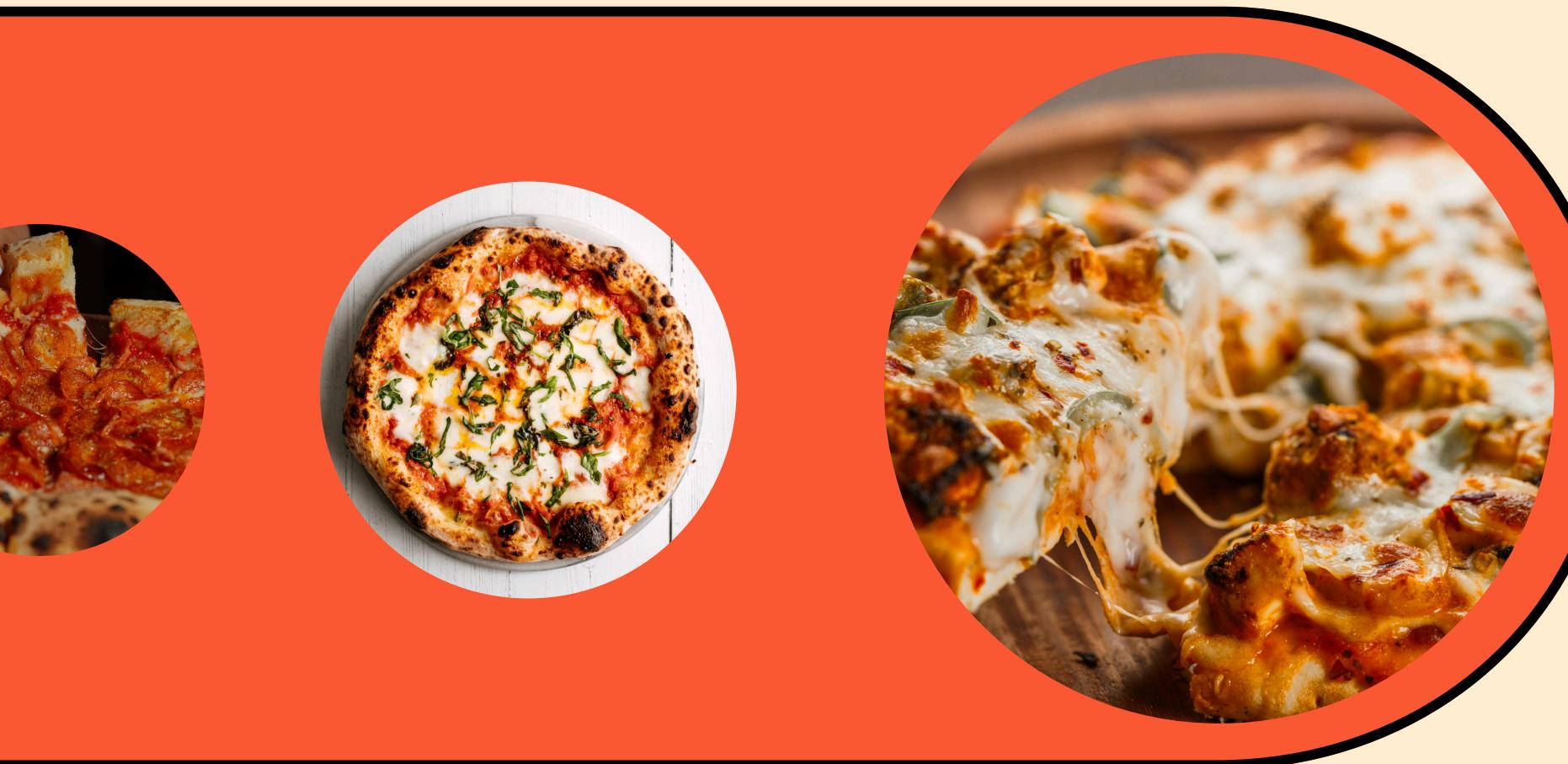


HELLO

MY NAME IS NAVEEN KUMAR. IN THIS PROJECT, I HAVE UTILIZED SQL QUERIES TO SOLVE THE QUESTIONS RELATED TO PIZZA SALES.



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



SELECT

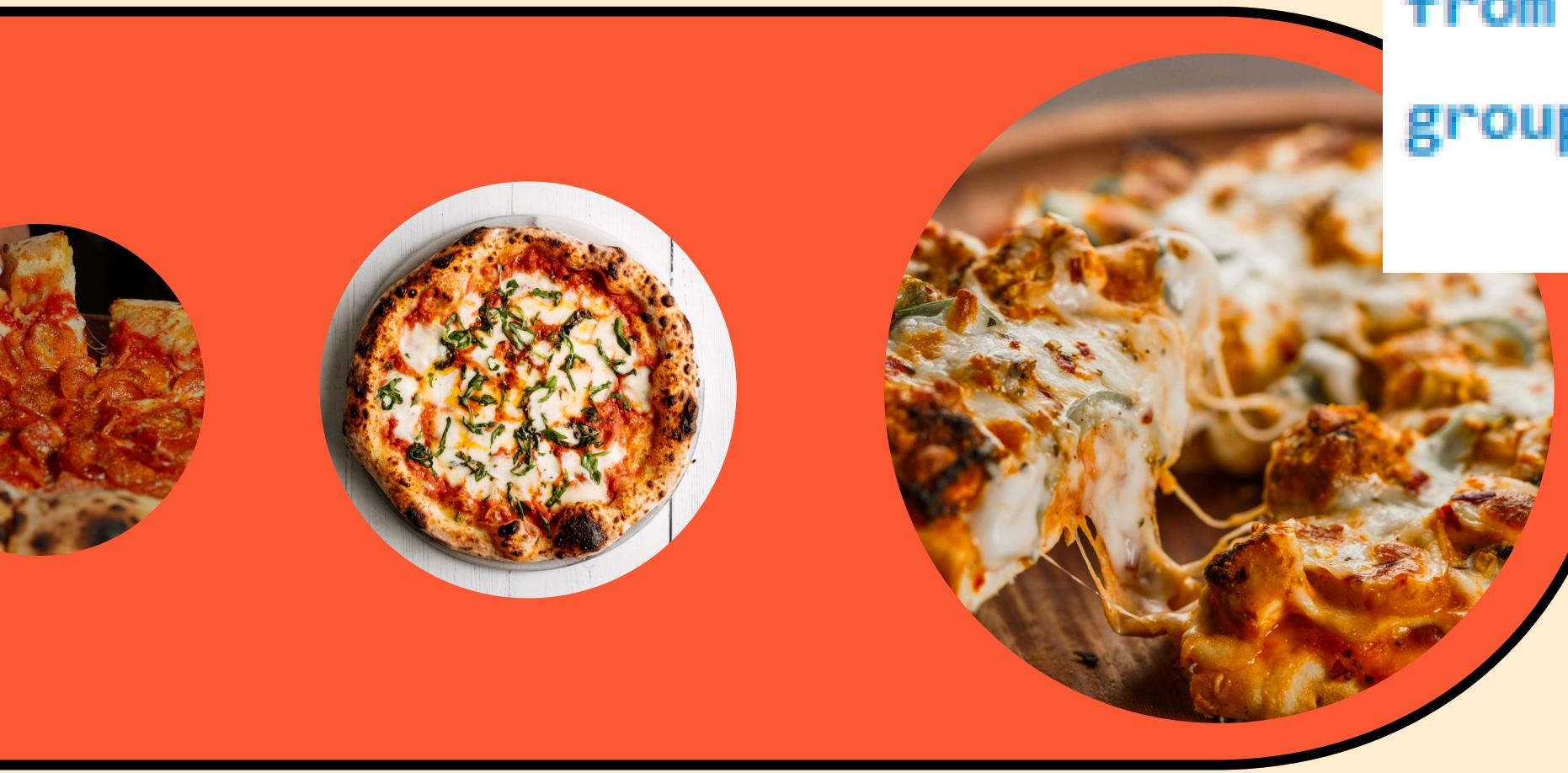
COUNT(order_id) AS total_orders

FROM

orders;

| Result Grid | |
|-------------|--------------|
| | total_orders |
| ▶ | 21350 |

Join relevant tables to find the category-wise distribution of pizzas.



```
select category, count(name) as pizzas  
from pizza_types  
group by category;
```

Result Grid |

| | category | pizzas |
|--|----------|--------|
| | Chicken | 6 |
| | Classic | 8 |
| | Supreme | 9 |
| | Veggie | 9 |

Calculate the total revenue generated from pizza sales.



Result Grid

| | total_revenue |
|---|---------------|
| ▶ | 817860.05 |

```
SELECT  
    ROUND(SUM(pizzas.price * order_details.quantity),  
          2) AS total_revenue  
FROM  
    pizzas  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id;
```





Determine the distribution of orders by hour of the day.

| hour | orders |
|------|--------|
| 11 | 1231 |
| 12 | 2520 |
| 13 | 2455 |
| 14 | 1472 |
| 15 | 1468 |

```
select hour(order_time) as hour,  
count(order_id) as orders  
from orders  
group by hour;
```

Identify the highest-priced pizza.

```
select pt.name,p.price  
from pizza_types as pt join pizzas as p  
on pt.pizza_type_id = p.pizza_type_id order by p.price desc limit 1;
```

Result Grid | Filter Rows:

| | name | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |



Determine the top 3 most ordered pizza types based on revenue.

```
select pt.name,sum(od.quantity*p.price) as revenue  
from pizza_types as pt join pizzas as p  
on pt.pizza_type_id = p.pizza_type_id  
join order_details as od  
on od.pizza_id = p.pizza_id  
group by pt.name  
order by revenue desc limit 3;
```

Result Grid | Filter Rows: Exp

| | name | revenue |
|---|------------------------------|----------|
| ▶ | The Thai Chicken Pizza | 43434.25 |
| | The Barbecue Chicken Pizza | 42768 |
| | The California Chicken Pizza | 41409.5 |



Identify the most common pizza size ordered.



```
select p.size, count(o.order_id) as ordered  
from pizzas as p join order_details as o  
on p.pizza_id = o.pizza_id  
group by p.size order by ordered desc
```

Result Grid | Filter

| | size | ordered |
|---|------|---------|
| ▶ | L | 18526 |
| | M | 15385 |
| | S | 14137 |
| | XL | 544 |
| | XXL | 28 |



Group the orders by date
-- and calculate the average number of pizzas ordered per day.

```
select avg(total_quantity)as total
> from(
select o.order_date ,sum(od.quantity)as total_quantity
from orders as o join order_details as od
on o.order_id  = od.order_details_id
group by  o.order_date
->)t
```

| | |
|---|----------------|
| | total |
| ▶ | 60.7709 |



List the top 5 most ordered pizza types along with their quantities.

```
select pt.name, sum(od.quantity) as total_quantity
from pizza_types as pt join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
on od.pizza_id = p.pizza_id
group by pt.name order by total_quantity desc limit 5;
```

| name | total_quantity |
|----------------------------|----------------|
| The Classic Deluxe Pizza | 2453 |
| The Barbecue Chicken Pizza | 2432 |
| The Hawaiian Pizza | 2422 |
| The Pepperoni Pizza | 2418 |
| The Thai Chicken Pizza | 2371 |





Join the necessary tables to find the total quantity of each pizza category ordered.

```
select pt.category,sum(od.quantity) as total_qumaturity  
from pizza_types as pt join pizzas as p  
on pt.pizza_type_id = p.pizza_type_id  
join order_details as od  
on od.pizza_id = p.pizza_id  
group by pt.category;
```

Result Grid | Filter Rows:

| | category | total_qumaturity |
|---|----------|------------------|
| ▶ | Classic | 14888 |
| | Veggie | 11649 |
| | Supreme | 11987 |
| | Chicken | 11050 |

Determine the top 3 most ordered pizza types based on revenue for each pizza category

```
with cte_revenue as (
  select pt.name,pt.category,
  sum(od.quantity * p.price) as revenue
  from pizza_types as pt join pizzas as p
  on pt.pizza_type_id = p.pizza_type_id
  join order_details as od
  on od.pizza_id = p.pizza_id
  group by pt.name,pt.category
),
cte_ranks as(
  select name,category, revenue,
  rank() over(partition by category order by revenue desc) as ranks
  from cte_revenue
)
select name,category,revenue from cte_ranks
where ranks <=3
)
```

| | name | category | revenue |
|---|------------------------------|----------|----------|
| . | The Thai Chicken Pizza | Chicken | 43434.25 |
| | The Barbecue Chicken Pizza | Chicken | 42768 |
| | The California Chicken Pizza | Chicken | 41409.5 |
| | The Classic Deluxe Pizza | Classic | 38180.5 |
| | The Hawaiian Pizza | Classic | 32273.25 |



Analyze the cumulative revenue generated over time.

```
select order_date,sum(revenue) over(order by order_date rows between unbounded preceding and current row)as cumulative_revenue from(  
select o.order_date,  
sum(od.quantity * p.price) as revenue  
from pizzas as p join order_details as od  
on p.pizza_id = od.pizza_id  
join orders as o  
on o.order_id = od.order_id  
group by o.order_date  
)t
```



| | order_date | cumulative_revenue |
|---|------------|--------------------|
| ▶ | 2015-01-01 | 2713.850000000004 |
| | 2015-01-02 | 5445.75 |
| | 2015-01-03 | 8108.15 |
| | 2015-01-04 | 9863.6 |
| | 2015-01-05 | 11929.55 |

Calculate the percentage contribution of each pizza type to total revenue

| pizza_type | percentage_contribution |
|---------------------------|-------------------------|
| The Hawaiian Pizza | 3.95% |
| The Classic Deluxe Pizza | 4.67% |
| The Five Cheese Pizza | 3.19% |
| The Italian Supreme Pizza | 4.09% |
| The Mexicana Pizza | 3.27% |

```
SELECT
    pt.name AS pizza_type,
    CONCAT(ROUND(SUM(p.price * od.quantity) / (SELECT
                                                SUM(p.price * od.quantity)
                                            FROM
                                                order_details AS od
                                            JOIN
                                                pizzas AS p ON p.pizza_id = od.pizza_id) * 100,
                2),
           '%') AS percentage_contribution
FROM
    pizza_types AS pt
    JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
    JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
```





THANK YOU